// 07 77   *3 / 74 9*

P. 5

FAST COMPUTATIONAL SCHEME OF IMAGE COMPRESSION FOR 32-BIT MICROPROCESSORS

Leonid Kasperovich
Space Research Institute
84/32 Profsoyuznaya St., Moscow, 117810, Russia
Fax: 7-095-310-7023

Abstract - This paper presents a new computational scheme of image compression based on the discrete cosine transform (DCT) , underlying JPEG and MPEG International Standards. The algorithm for the 2-d DCT computation uses integer operations (register shifts and additions / subtractions only), its computational complexity is about 8 additions per image pixel. As a meaningful example of an on-board image compression application we consider the software implementation of the algorithm for the Mars Rover (Marsokhod, in Russian) imaging system being developed as a part of Mars-96 International Space Project. It's shown that fast software solution for 32-bit microprocessors may complete with the DCT-based image compression hardware.

INTRODUCTION

The discrete cosine transform (DCT) is widely applied in various fields including image data compression and was chosen as a basis of International JPEG (Joint Photographic Experts Group) and MPEG (Motion Pictures Experts Group) image / video Compression Standards. The DCT technique is applicable to the digital representations of natural scenes and other types of continuous tone gray-scale and color images.

An extensive research experience in the field of DCT studying has been summarized in the various publications and textbooks (e.g., Pennebaker et al., 1993). The most meaningful example of the 8x8 DCT implementation (Feig et al., 1992) uses 94 real multiplications and 454 additions, but only 54 multiplications and 462 additions in a scaled version, where the DCT computation is followed by normalizing and quantization.

Due to the rounding-off and truncation effects of the quantization process in image compression, one can carry out , in practice, all DCT calculations approximately, not increasing the overall computational error. Then making use of the floating-point multiplications is not necessarily. In this way, a technique based on generalized Chen transform for approximating with rational numbers the scales 8x8 DCT, has been developed that uses 608 additions per 8x8 image fragment (Allen et al., 1992).

This paper presents an improved algorithm on the basis of the scaled 8x8 DCT approximation method that has been previously published by the author in cooperation with Dr.V.F.Babkin (Kasperovich et al., 1993). The algorithm presented uses 530 additions (vs. 684 as before) per 8x8 block that is a little bit more than the overall number of arithmetical operations used in the Feig-Winograd algorithm, but considerably fewer than in the approximation algorithm by Allen and Bronstein.

Note that in the wide range of microprocessors a floating-point multiply execution takes ordinarily more processor clock cycles than a summation of integers, hence a

multiplication-free algorithm might be preferable in the applications. This paper consists of 3 sections discussing the algorithm, its accuracy and on-board implementation performance.

DCT DECOMPOSITION

The two-dimensional forward DCT (FDCT) of an input 8x8 block consisting of integers $X_{i,j}$ $i = 0, 1, ..., 7,$ $j = 0, 1, ..., 7$ is defined by the following formula:

$$Y_{m,n} = \frac{1}{4} K(m) K(n) \sum_{i=0}^{7} \sum_{j=0}^{7} X_{i,j} \cos \frac{(2i+1)m\pi}{16} \cos \frac{(2j+1)n\pi}{16}$$

where:

$$K(t) = \begin{cases} 1/\sqrt{2}, t = 0 \\ 1, otherwise \end{cases}$$

$m = 0, 1, ..., 7,$ $n = 0, 1, ...7.$

The FDCT can be accomplished in row-column fashion using one-dimensional transform:

$$
\begin{pmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \\ Y(6) \\ Y(7) \end{pmatrix} = \frac{1}{2}
\begin{pmatrix}
\gamma(4) & \gamma(4) & \gamma(4) & \gamma(4) & \gamma(4) & \gamma(4) & \gamma(4) & \gamma(4) \\
\gamma(1) & \gamma(3) & \gamma(5) & \gamma(7) & -\gamma(7) & -\gamma(5) & -\gamma(3) & -\gamma(1) \\
\gamma(2) & \gamma(6) & -\gamma(6) & -\gamma(2) & -\gamma(2) & -\gamma(6) & \gamma(6) & \gamma(2) \\
\gamma(3) & -\gamma(7) & -\gamma(1) & -\gamma(5) & \gamma(5) & \gamma(1) & \gamma(7) & -\gamma(3) \\
\gamma(4) & -\gamma(4) & -\gamma(4) & \gamma(4) & \gamma(4) & -\gamma(4) & -\gamma(4) & \gamma(4) \\
\gamma(5) & -\gamma(1) & \gamma(7) & \gamma(3) & -\gamma(3) & -\gamma(7) & \gamma(1) & -\gamma(5) \\
\gamma(6) & -\gamma(2) & \gamma(2) & -\gamma(6) & -\gamma(6) & \gamma(2) & -\gamma(2) & \gamma(6) \\
\gamma(7) & -\gamma(5) & \gamma(3) & -\gamma(1) & -\gamma(1) & -\gamma(3) & \gamma(5) & -\gamma(7)
\end{pmatrix}
\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{pmatrix}
$$

where: $\gamma(k) = \cos(2\pi k / 32)$.

Setting $\Psi = \sqrt{2}$, $C_1 = \gamma(1)/\gamma(7)$, $C_2 \Psi = \gamma(3)/\gamma(7)$, $C_3 \Psi = \gamma(5)/\gamma(7)$ and representing the transformed values $Y(i)$ in a "quasi-complex" form $R(i) + \Psi A(i)$, leads to the Kasperovich - Babkin FDCT algorithm mentioned above, in which the attends in the formula for the 2-d FDCT values $[R(R) + 2A(A)] + [A(R) + R(A)]\sqrt{2}$ are represented through the "basic" elements $A(A)$ by means of additions and subtractions. 64 multiplications by the

constants C_1, C_2, C_3, which are closed to 5,3 an 2, are sufficient for obtaining the basic elements. All multiply operations by these 3 constants are substituted in the DCT approximation by the additions and subtractions. Further,

$$2X_1 = [X_3 + X_5 \Psi]^*$$

$$X_2 = (1 + \Psi) X_6^*$$

$$X_3 \Psi = (X_1 + X_7)^*$$

where: $(a + \Psi b)^* = a - \Psi b$

Thus, only 30 of 60 multiplications by $\sqrt{2}$ should be computed, which are practically replaced

with a Taybor series approximation: $\sqrt{2} \approx 1 + \dfrac{1}{2} - \dfrac{1}{16}$.

Theorem. i) FDCT can be performed as an operator composition $FDCT = F \circ D \circ C \circ T$, where

      T is a preliminary transform (192 preadditions),

      C - computation of the basic elements,

      D - deriving the output values,

      F - pointwise factorization (scaling);

      ii) C uses 64 multiplications by predefined constants, D calls 30 multiplications by $\sqrt{2}$;

      iii) Approximation of C uses 144 additions, approximation of D calls 194 additions;

      iv) $IDCT = T^t \circ F \circ D \circ C$;

The transformations T and C are separable (i.e. can be computed in row-column fashion□) meanwhile D is non-separable 2-d transform. Generally speaking, the number of preadditions equals to 224 (as mach as in Feig-Winograd algorithm), but the certain part of it is done while computing the basic elements (C transformation) in order to preserve the algorithm symmetry.

32-BIT IMPLEMENTATION

The DCT itself is parallelizable that makes it possible to group data elements in such a way, that DCT computation could be considered as sequential single-instruction/multiple-data process. In particular, two additions $a+b$ and $c+d$ can be achieved in one _(a,c) + (b,d)_, coupling the elements of an input 8x8 block into the pairs. Assuming that all computations can be done with 16-bit arithmetic, that observation is applicable to a single microprocessor taking the substantial advantages of a full-length processor word of 32-bit or newest 64-bit devices.

Since the image data precision is ordinarily 8-bit per sample and the average number of summation per point is 530/64 < 8.3 in our algorithm, then in most case (48 of 64) the computations are done within 16-bit range and can be paralleled as mentioned above. However, this is worthy in a case of multiplication-free computational scheme, because a

fractional multiplying will destroy the least significant 16-bit word of a pair: .In turn, an additional error in most significant 16-bit word produced in our algorithm by a carry bit of addition/subtraction of the least significant 16-bit words can be neglected due to the scaling performed by the operator F and quantization.

The test of *LENA* standard image gives a good illustration of the tolerable computational accuracy, comparing the algorithm presented with a direct floating-point method. The maximum pint-size difference between the original and expanded pictures is identical for both methods, the mean arithmetic modules error is slightly different: 3.516 versus 3.501 in the direct computation.

APPLICATION TO THE ON-BOARD PROCESSING

In this section we consider the Mars Rover imaging system, that contains a panoramic camera along with 2 stereo cameras. Three compression modes are planned:

•Receiving the descent camera images compressed as the separate frames (specified data rate is 1 frame of size 512x512x8 bit per second).

•Compression of high resolution panoramic camera still images.

•Image sequence compression to create the virtual environment from real Martian surface data in order to control and navigate the rover manually.

In this way, an image compression module (ICM) based on JPEG compression chip set from Matra Marconi Space (France) was supposed to be installed in Mars Rover as a hardware accelerator board. The ICM technical specifications are 3 watts consumption at 1 megapixel /sec; 12000 mm; 200 grams (see Mars-94 in the pictures, 1992). The chip set contains the two CMOS ASICs.

An alternative approach implementing in software a new algorithm to compute the DCT in multiplication-free 32-bit arithmetic seems to be more preferable. In order to provide the autonomy of movement, control and timing experiments, data collection and storing etc., the rover is equipped with a on-board computer based on the powerful 32-bit T805 transputer from INMOS Corporation (see Transputer Data Book, 1990), that can be regarded both as a special (i.e. image processing) and a general purpose processor. Major characteristics of IMS-T805 are:

•32 bit internal and external architecture.

•30 MIPS (peak) instruction rate.

•4 Kbyte on-chip RAM direct addressable.

•Internal timers.

•4 fast Serial Links (10 Mbit/sec).

•Less than 1 watt power consumption at 30 Mhz.

The heart transputer modules, which are the real copy of each other both electrically and even mechanically. There is no distinguished one among them as far as the access to the peripheral blocks concerned, but, and it is a substantial point, only two out of four transputer modules are powered at a time. Which two, it is determined by the actual state of the overswitch logic (Balazs et al., 1994).

The software implementation of the image compression algorithm for the on-board computer provides the same compression rate as ICM hardware, requiring no additional

weight and power consumption. Compression mode 3 gives a good illustration of the software solution flexibility, where DCT computation for intra- and interframe compression is combined with another algorithm (Motion Estimation) for the successive frame matching, that is a part of stereo-based autonomous navigation software.

CONCLUSION

The reliability and performance of the Mars Rover systems including on-board computer and the application software have been evaluated in the several tests with the real test site observation (e.g. Kamchatka, Far East, Russia, August 1993 and Mohave Desert, California, US, March 1994). The rover control as well as the compressed data transmission has been provided via satellite communication link. The results are quite good and show the possibility to use the software solution of the special tasks in various applications, in particular image processing and compression, where hardware assistance is currently required

REFERENCES

Pennebaker, E., & Winograd, S. (September 1992). Fast algorithms for the discrete cosine transform. *IEEE Trans. Signal Processing, 40*(9), 2174-2193.

Allen, J.D., & Blonsein, S. (July 1992). US patent # 5, 129,015.

Kasperovich, L.V., & Babkin, V.F. (1993). Fast discrete cosine transform approximation for JPEG image compression. *Lecture Notes in Comp. Sci., 719,* 98-104.

Mars-94 in the picture. (July 1992). *News from Prospace, 34,* 48-50.

Balazs, A. & Biro, J. & Szalai, S. (15-21 May 1994). Onboard computer for Mars-96 rover. *Proc. of 2-nd Int. Sympos. on Missions, Technologies and Design of Planetary Rovers.* Center National d'Efudes Spatiales(CNES, France)-Russian Space Agency (RKA), Moscow-St.-Petersburg (Russia) (to be published).

Transputer Data Book. (1992). 2 nd Edition, London, UK:Prentice Hall.